Jean-Marie Culot

GUIDE DE RÉFÉRENCE

Apache 2

© Éditions OEM (Groupe Eyrolles), 2003,
ISBN 2-7464-0490-7

OEM

EYROLLES

Chapitre 14

Installer et configurer PHP

PHP est un langage de développement très répandu, utilisé dans une large gamme de serveurs, dont Apache. Nous allons donc prendre connaissance de ses principales caractéristiques, puis évaluer leur intérêt dans la mise en œuvre d'un serveur Apache.

Sa force tient notamment à sa capacité de gérer les accès à des bases de données, de supporter la création graphique, de produire du code HTML...

Dans ce chapitre, nous allons apprendre :

- ce qu'est PHP et comment il travaille ;
- comment l'installer ;
- en quoi consistent ses fonctionnalités.

Introduction à PHP

PHP est un langage de développement, c'est-à-dire qu'il permet de rédiger des programmes ! En voici un exemple, assez simple, on peut en convenir :

```
< ? echo "Bonjour, la compagnie " ;  ?>
```

L'instruction echo va diriger son complément « Bonjour, la compagnie » vers la sortie standard. Et puisque l'instruction sera placée dans le serveur, la sortie standard sera la réponse du serveur au client. L'utilisateur verra donc apparaître à son écran : « Bonjour, la compagnie ».

Dans le monde des langages Perl, PHP, Java, etc. on réserve le nom de programme aux objets dits « compilés » (p.ex. les exécutables suffixés par *.exe* ou à leurs extensions de type .dll). Et on parle de scripts pour les textes rédigés par les réalisateurs de sites, textes qui sont en fait une liste d'appels (on dira une liste d'instructions) aux fonctions offertes par le programme.

Notre texte à la page précédente est donc un script : il appelle un affichage à l'écran via l'instruction echo, et activera le programme php.exe qui en assurera la validation, l'interprétation, la compilation et l'exécution.

Des traitements par scripts

PHP est orienté vers la rédaction de scripts destinés à être placés dans le serveur, et exécutés par lui : « *server-side* », à l'inverse d'autres langages où l'exécution est reportée sur le *browser*. Il est apparenté au JSP (*JavaServer Pages*) de Sun et à ASP (*Active Server Pages*) de Microsoft. Il est placé dans Apache2.0 en tant que filtre, mais un filtre qui, en plus de filtrer, modifie ce qui le traverse : il reçoit des pages comportant du code PHP, interprète cette partie de code, génère un texte html imbriqué et fournit en sortie une page complétée, en format html, à destination du client.

Un exemple

Une page élémentaire de PHP pourrait donc ressembler à ceci :

```
<h1> < ? echo "Bonjour, la compagnie " ; ?>  </h1>
```

ou encore

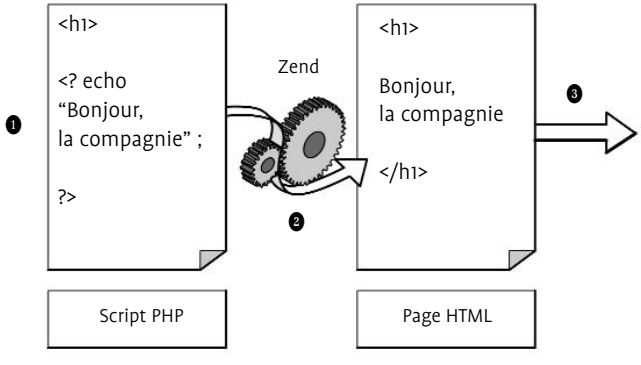
```
<h1> Bonjour < ? $quelquechose = ", la compagnie " ; echo $quelquechose ; ?>
!</h1>
```

Vous reconnaissez d'abord le cadre d'une page HTML avec ses balises de début <h1> et de fin </h1>. À l'intérieur, trois éléments : le texte « Bonjour », un ensemble cadré dans des balises < ? ?>, et le point d'exclamation.

Le texte « Bonjour » et le point d'interrogation sont lisibles en tant que texte HTML par le *browser*. Entre les balises < ? et ?> se trouve un texte PHP qui doit d'abord être interprété par le module PHP pour devenir du texte HTML.

Un autre exemple

Dans notre second exemple, la commande echo affichera le contenu de la variable $quelquechose. Le *browser* affichera ainsi « Bonjour, la compagnie ! ». C'est donc bien dans le serveur que le texte a été traité par PHP, et c'est le résultat qui est envoyé au *browser*, ce dernier ne voyant que du HTML. Dans l'incapacité de reconstituer le processus de création de la page, il ne peut donc connaître le script PHP concerné.



❶ Un script, rédigé par le développeur grâce à un éditeur ascii quelconque, est placé dans un répertoire du site.

❷ Lorsque ce document est appelé par le client, Apache le confie au programme PHP.exe qui, grâce au moteur Zend, interprète le script et, en l'exécutant, produit un résultat HTML.

❸ La page HTML ainsi générée est envoyée au browser du client qui pourra l'afficher.

L'exemple est élémentaire. Mais vous devinez les possibilités offertes par cette aptitude à produire des réponses, non pas seulement à partir de documents fixes, mais en activant des procédures, en interrogeant des bases de données. On parlera à ce sujet de texte généré « à la volée ».

Les demandes de vos clients pourront être honorées avec un bon niveau de qualité, niveau apprécié et désormais indispensable. Notamment, l'information qu'ils recevront en retour sera obtenue à partir de bases de données tenues à jour.

Un peu d'histoire

Les débuts

Le concepteur de ce langage est Rasmus Lerdorf. C'était... il y a très longtemps, en 1994. Rasmus cherchait à vérifier qui venait consulter son CV en ligne. La première version publique fut disponible au début de l'année 1995. C'était un analyseur de texte extrêmement simple, assorti d'utilitaires couramment employés sur les pages Web, un livre d'or, un compteur, ... Bientôt il y ajouta un interpréteur de formulaires HTML et le support de mySQL. C'est comme cela que naquit PHP. Il grandit de manière spectaculaire et de nombreuses personnes contribuèrent à son amélioration.

Durant l'été 1997, 50 000 sites Web avaient déjà adopté PHP. C'est à ce moment que d'un projet personnel, on passa à un projet d'équipe. L'analyseur fut réécrit par Zeev Suraski et Andi Gutmans et bientôt dénommé ZEND. Il forma le support de PHP3.

La version courante à ce jour est PHP4. Une nouvelle génération est en chantier, visant l'amélioration des performances et le support d'une gamme plus large de serveurs autres que Apache.

Aujourd'hui, PHP est disponible gratuitement selon les licences de l'OpenSource. Vous trouverez en *www.php.net* le produit lui-même (sous forme source et compilée pour différents systèmes d'exploitation), mais aussi toute la documentation sur le produit comme sur le projet. Ainsi qu'une invitation à collaborer au projet...

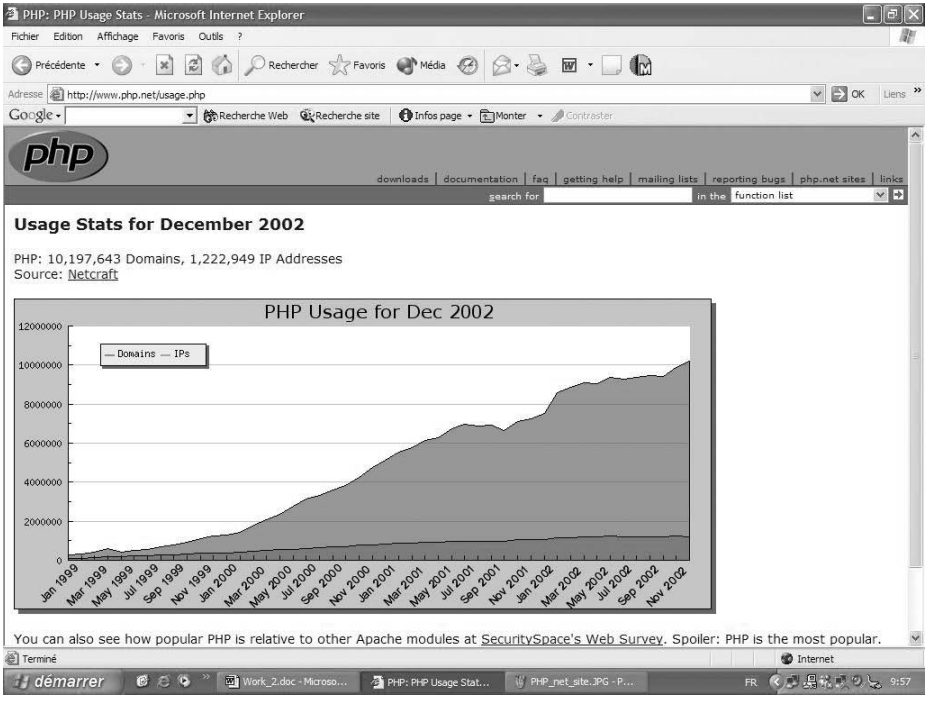
Le succès

Le succès est considérable. On estime qu'il est implanté sur plus de sept millions de sites et qu'il a été retenu pour habiller plus de la moitié des serveurs Apache, ce qui en fait le module de développement le plus prisé.

Pour connaître la pénétration du marché acquise par PHP par rapport à d'autres langages, vous pouvez consulter les chiffres tenus à jour par Netcraft Survey :

http://www.securityspace.com/survey/

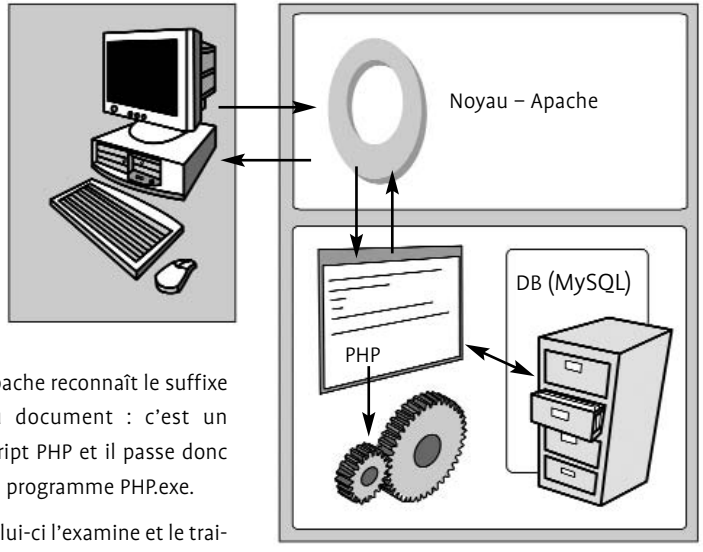
En décembre 2002, sur les 8 millions et demi de serveurs observés par Netcraft, 62% étaient construits sur Apache (5 329 636). Dans le seul domaine des serveurs Apache, plus de 2 500 000 modules PHP étaient installés, présents donc sur près d'un serveur sur deux.


PHP: PHP Usage Stats - Microsoft Internet Explorer
Usage Stats for December 2002
PHP: 10,197,643 Domains, 1,222,949 IP Addresses
Source: Netcraft
PHP Usage for Dec 2002
Domains
IPs
You can also see how popular PHP is relative to other Apache modules at SecuritySpace's Web Survey. Spoiler: PHP is the most popular.

Aujourd'hui, PHP (en versions 3 et 4) est inclus dans de nombreuses distributions, notamment par les distributeurs de produits et services OpenSource, tels que RedHat, SuSe,...

Vous trouverez en librairie une profusion d'ouvrages sur PHP ; deux d'entre eux vous seront présentés en fin de chapitre.

La demande du client concerne une fiche de la base de données



❶ Apache reconnaît le suffixe du document : c'est un script PHP et il passe donc au programme PHP.exe.

❷ Celui-ci l'examine et le traite grâce au moteur Zend : il fait appel à une des extensions, MySQL, qui extrait la fiche de la DB.

❸ Le programme PHP formate la réponse en HTML et la passe à Apache qui l'achemine vers le client.

L'architecture de PHP

PHP fonctionne grâce à un moteur d'interprétation et d'exécution nommé ZEND et aux extensions de ce moteur connectées via des API (Interfaces d'applications). Beaucoup de ces extensions sont livrées à l'intérieur de librairies C et donc disponibles pour les développeurs : ainsi, les connexions à des bases de données, à des répertoires LDAP ; les connexions à des serveurs de mail ; les créations et manipulations d'objets graphiques.

Le serveur SAPI (*ServerAPI*) fournit une interface entre tous les serveur Web, dont Apache et PHP.

Les avantages de PHP

Facilité d'apprentissage

Ce langage est tout désigné pour qui souhaite intégrer à son serveur un système de création dynamique de contenu, et qui ne dispose pas d'une culture approfondie de programmation. PHP se prête bien à un apprentissage graduel en incluant, dans les pages HTML existantes, des blocs de texte PHP très simples, puis progressivement plus complexes.

Grande satisfaction que de pouvoir tester aisément et rapidement des extensions à votre site.

Contexte Open Source

PHP est distribué sous une licence de même type qu'Apache, autorisant une intégration dans des projets commerciaux ou non. Un réseau mondial de talentueux développeurs travaille sans cesse à l'évolution de ce langage : c'est donc un type de produit qui est vérifié, testé et maintenu en ordre bien plus que beaucoup de produits commerciaux. Il évolue donc au fur et à mesure des besoins présentés par une multitude de collaborateurs, et ses évolutions sont le fruit d'un large consensus.

Large assistance

PHP est très répandu. Il est ainsi aisé de trouver des programmeurs ayant une bonne expertise du sujet, qui ont commenté les facettes du produit et se prêtent à répondre aux questions. Il n'y a qu'à voir la profusion de sites, de news, de FAQS, de listes de mails disponibles sur le Net.

Dans l'esprit Open Source, une belle panoplie d'extensions prêtes à l'emploi et d'exemples de codes est à libre disposition pour des objectifs divers, depuis la manipulation de XML jusqu'aux accès aux répertoires et bases.

Support de bases de données

PHP offre la connectivité à ODBC, à des bases de données Open Source telles que MySql, PostgreSQL, ainsi qu'à des bases commerciales comme Oracle, DB2,...

Support multi-plates-formes

Bien qu'issu du monde Unix, PHP tourne sur une large gamme de systèmes d'exploitation tels que Windows, MacOS, OS/2,... Il peut être intégré à un grand nombre de serveurs Web, Apache en Open Source, mais aussi aux serveurs de Microsoft et de Netscape. Ceci permet aux réalisateurs de sites de développer des solutions pour une combinaison particulière de ser-

veur/base de données/système d'exploitation, puis de les proposer sur des combinaisons différentes.

Contextes sécurisés

Les fournisseurs d'accès au Web et d'hébergement peuvent proposer PHP à leurs clients en leur assurant une complète imperméabilité de leur site vis-à-vis des voisins ou d'intrus, même si l'infrastructure est partagée.

Maintenance aisée

PHP est compilé avant exécution. Cette étape est complètement transparente au développeur, ce qui lui permet de procéder à ses modifications et de les vérifier tout aussitôt sur le *browser*. Par comparaison, le cycle de développement/compilation/test est plus long en contexte Java.

Support commercial

Beaucoup de sociétés proposant des services en Internet considèrent que PHP fait partie des composants critiques des solutions, et ont développé des compétences en ce domaine. Ainsi des clients projetant d'installer des sites d'entreprise peuvent-ils être assurés de trouver sur le marché des compétences en PHP pour le développement comme pour le suivi.

Et les inconvénients

Au moment de l'option, vous aurez aussi à prendre en compte les désavantages.

Une maintenance hasardeuse

PHP peut se trouver victime de sa facilité d'implantation. En l'absence d'une analyse et d'une organisation adéquates, la maintenance risque de virer au cauchemar. Cela ne tient pas à PHP, mais à l'aptitude à le mettre en œuvre à l'intérieur d'un projet cohérent.

Un surdimensionnement

PHP a intégré des fonctions avancées, telles que celles proposées par des langages orientés objet. Ces possibilités s'avèrent démesurées pour un grand nombre de sites et peuvent encombrer leurs développeurs.

Une cohérence à vérifier

Un grand nombre d'évolutions sont constamment en chantier, et à des niveaux différents de réalisation. Le moment venu, elles sont incorporées dans la version courante de PHP, mais elles auront à être couplées avec des interfaces de bases de données fournies par ailleurs. Il peut être fastidieux de vérifier soi-même les compatibilités. Ceci explique que beaucoup de fournisseurs présentent PHP dans des solutions prêtes à l'emploi, comportant serveur et base de données.

Une réticence de la part des entreprises

PHP souffre des mêmes a priori que d'autres produits *OpenSource* en milieu commercial. Bien qu'il soit de haute qualité et très répandu en milieu *OpenSource*, et qu'il fasse ses preuves dans un nombre impressionnant de serveurs de par le monde, sa visibilité en entreprise est moindre que celle de Java (*Java Server Pages / servlets*) ou ASP, et la démonstration de son intérêt exigera plus d'effort.

Installation sous Win32

Installation sur un serveur préexistant

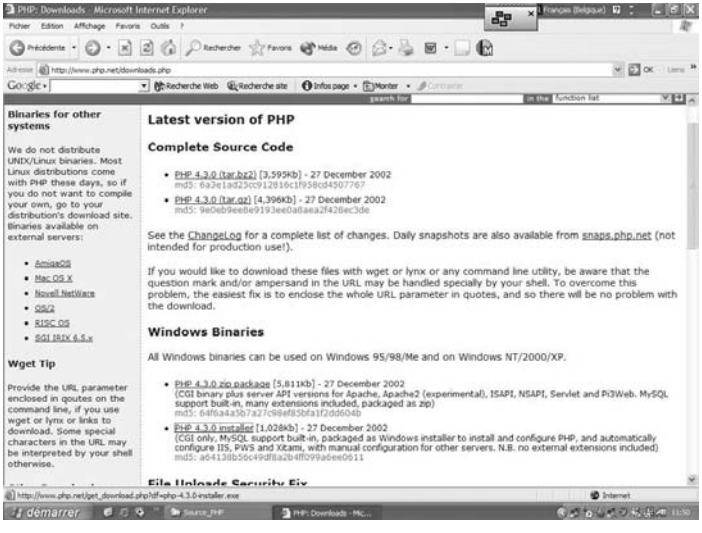
Une configuration prête à l'emploi a été réalisée sous le nom de EasyPHP et vous permet d'installer très facilement, et en une seule procédure, les 3 composants majeurs d'un site : un serveur Web, un générateur de pages html et un gestionnaire de bases de données. Les produits retenus relèvent du monde de l'*OpenSource* : Apache, PHP et MySql.

C'est une formule que nous aurions pu préconiser dès le début de notre itinéraire : elle offre une solution complète, et très aisément. Nous la présenterons brièvement plus bas.

Mais nous choisissons de procéder pas à pas. Et c'est sur un serveur HTTP déjà installé que nous allons placer PHP. Et c'est ce seul produit que nous irons donc chercher sur le Web, décharger et installer. Si vous vérifiez le fichier de configuration d'Apache, dans sa version initiale, vous ne trouverez pas mention de PHP.

PHP associé ou imbriqué ? PHP peut fonctionner avec Apache de deux manières différentes :

1. soit rester **un programme indépendant** dont Apache demande la collaboration via le procédé CGI (*Common Gateway Interface*) présenté au chapitre précédent. PHP peut d'ailleurs fonctionner de cette manière avec tout serveur Web ;
2. soit se présenter simplement comme **un des modules Apache**, au même titre que d'autres ; PHP ne constitue plus alors qu'un composant d'un ensemble, cet ensemble étant compilé (traduit en langage machine) : les performances sont améliorées, et sur un site de production, ce serait l'option à privilégier.



Installation en tant que programme externe sous CGI

C'est la première manière que nous allons appliquer ici pour sa facilité d'installation, fidèles que nous sommes à procéder pas à pas, et du plus simple au plus complexe.

Les développeurs de PHP présentent en effet une version « CGI » accompagnée d'un installateur qui va nous servir de guide. Rendez-vous sur le site de PHP : *www.php.net* et, après un rapide tour de reconnaissance, rendez-vous sur la page de téléchargement...

À ce jour, c'est la version 4.3.0 qui est préconisée.

Le choix du code source

Un premier paragraphe vous propose le code source complet de PHP, ce qui est bien dans l'esprit de l'*OpenSource*. Vous en disposez donc pour le retravailler à votre guise si l'inspiration vous vient.

Vous en disposez aussi si vous optez pour une installation sous Windows ou sous Linux que vous effectueriez avec compilation.

- Nous choisissons PHP déjà compilé, sous forme de « *binary* ». Nous ne sommes pas intéressés par les « *patches* » qui améliorent des versions antérieures au 4.3.0.
- Nous optons pour le produit auquel est associé un installateur Windows : PHP 4.3.0 Installer. Le commentaire vous annonce qu'il s'agit bien de la version CGI, qu'il est prêt à être installé avec l'utilitaire Windows, et qu'il assurera dans la foulée la configuration de plusieurs serveurs : IIS, PWS,...

Mais vous n'aurez pas trouvé Apache, ni Apache2 dans la liste des serveurs configurés d'emblée. Ce qui signifie que l'installateur ne nous aidera que pour PHP même, et que nous aurons à intervenir pour expliquer à Apache comment collaborer avec PHP.

Cette section s'applique aux systèmes dits « Win32 » : Windows 95/98/ME *et* Windows NT/2000/XP avec, à ce jour, une réserve pour XP où la configuration de PHP est toujours considérée au stade du test.

Installation du programme

Après téléchargement dans un répertoire de travail :

- double-cliquez sur l'objet obtenu : PHP-4.3.0-installer.exe et suivez les instructions ;
- pour le répertoire cible de votre installation, choisissez librement. Mais, pour notre pas à pas, acceptez la proposition de C:\PHP ;

- pour le paramétrage de SMTP, le composant du serveur capable de traiter le mailing, acceptez aussi les valeurs proposées. Dans notre itinéraire, nous ne ferons pas référence à cette fonctionnalité.

À ce jour, l'installateur n'était pas configuré pour installer aussi Apache ; il dit le regretter et vous invite à procéder à cette installation manuellement.

- Rendez-vous en C:\PHP.

Nous y trouvons, en plus des notes d'installation :

- le php.exe : c'est bien le programme compilé, destiné à collaborer avec Apache ;
- le fichier .ini sous forme php.ini-dist à moins que l'installateur ne l'ait déjà placé en contexte system.
- Réalisez une copie de ce fichier php.ini-dist et conservez-la telle quelle afin de pouvoir y réaccéder pour vérifier ce qu'étaient les paramètres par défaut. Renommez la copie : php.ini.
- Déplacez-la en contexte système :
  - en C:\Windows pour Windows 95/98
  - en C:\Winnt pour Windows NT/2000/XP

Et ouvrez-la avec un éditeur de texte quelconque.

Localisation des scripts

Le programme php.exe est prêt à travailler, mais il faut à tout le moins lui indiquer à quel endroit vous allez placer les pages et les scripts qu'il aura à traiter. Il ne peut pas deviner dans quel répertoire vous les aurez placés. Il est logique qu'ils se trouvent dans votre site, encore faut-il indiquer à PHP où chercher celui-ci et les pages qui lui seront soumises.

Recherchez le paramètre « doc_root » ; indiquez-y le répertoire de votre site ou le répertoire particulier dédié aux pages PHP. Ce répertoire fera l'objet de mesures de sécurité, et les pages et scripts de PHP y seront protégés de même.

Choisissons, pour respecter nos options antérieures : C:\Program Files\Group Apache\Apache2\htdocs.

Une seconde modification du php.ini préparera la configuration des extensions (les php_*.dll) que l'on pourra fournir à PHP, et qu'il chargera à la demande : base de données, traitement de l'image... Comme vous les placerez vraisemblablement en C:\PHP, donnez ce répertoire comme paramètre à extension_dir.

À ce point, les autres paramètres indispensables étant garnis par défaut, le noyau de PHP est prêt à fonctionner. Il est en mesure d'interpréter un texte aussi sympathique que :

```
< ? echo "Bonjour, la compagnie " ;  ?>
```

C'est un début. Rien ne vous empêchera plus tard de faire naviguer PHP dans de superbes bases de données, ou de contrôler des flux monétaires... en lui fournissant quelques indications supplémentaires.

Annoncer PHP à Apache

Mais Apache ignore ce nouveau collaborateur. Nous allons le lui présenter.

- Ouvrons le fichier de configuration httpd.conf.

Pas de mention de PHP jusqu'à présent. Il pourrait ne jamais y en avoir si vous envisagez de produire des pages dynamiques avec des langages tels que Perl ou Java plutôt que PHP ou si vous n'avez pas de pages dynamiques à générer.

Nous avons 3 indications à fournir à Apache. Plaçons-les à un endroit quelconque du fichier mais, par cohérence, près d'indications du même type.

Lorsqu'on avertit Apache qu'il doit fonctionner en mode CGI, que des scripts seront placés en bonne place pour qu'ils soient traités par des programmes ou pas simplement considérés comme documents, on le lui indique par la directive ScriptAlias.

Une telle indication figure déjà dans le httpd.conf :

```
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
#
ScriptAlias /cgi-bin/ "C:/Program Files/Apache Group/Apache2/cgi-bin/"
```

Nous signalons à Apache que PHP doit être considéré comme programme CGI, et qu'il peut lui confier des scripts à exécuter.

```
ScriptAlias /php/ "C:/php/"
```

Nous connaissons la convention qui consiste à suffixer des documents pour qu'Apache reconnaisse dans le suffixe le type de traitement qu'il doit lui appliquer. Nous avons donc à lui dire de ne pas être étonné de trouver désormais des documents suffixés .PHP. Et nous lui précisons quelle application doit traiter ce type MIME particulier.

Plaçons la directive en compagnie des autres AddType

```
AddType application/x-tar .tgz
AddType image/x-icon .ico
#
# ----- Ajout de directive lors de l'ajout de php en tant que module cgi
AddType application/x-httpd-php .php
# ----- Fin ajout de directive lors de l'ajout de php en tant que module cgi
#
```

Et ce type d'application désormais sollicité, par quoi va-t-il être représenté ?

```
# Action lets you define media types that will execute a script whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#
# ----- Ajout de directive lors de l'ajout de php en tant que module cgi
Action application/x-httpd-php "/php/php.exe"
# ----- Fin ajout de directive lors de l'ajout de php en tant que module cgi
```

Notez la barre oblique requise dans le fichier de configuration d'Apache, même lorsqu'il s'agit de désigner des chemins d'accès en Windows.

- Sauvez votre fichier de configuration.
- Redémarrez (*Restart*) Apache.

Vous êtes impatient... et prêt à tester PHP.

- Une installation de ce type est indiquée pour une prise de connaissance et une première mise en œuvre de PHP « en un tour de main », ce qui est notre propos. Elle n'est cependant pas suffisamment sécurisée, méthode qui impliquerait une lecture approfondie de toute la documentation et un choix approprié des options. Ce que l'on attendrait d'un serveur de production comportant des exigences de prévention.

Installation en tant que module Apache

Une seconde manière d'installer PHP sur Windows est de procéder manuellement.

La version que vous téléchargez est alors celle du package à décompacter, si vous voulez disposer d'emblée du package compilé : Php_.4.3.0_zip_package.

Un guide a été rédigé par Bob Silva et est disponible en version originale à l'adresse suivante :

- *http://www.umesd.k12.or.us.php/win32install.html*

Il assiste les installations sur différents serveurs, dont Apache, IIS... Il donne les indications aussi bien pour une installation à partir d'un package d'objets compilés qu'à partir des sources. Et dans ce second cas, vous donne les indications de compilation sous Windows.

Pour un gestionnaire de site déjà expérimenté, il y a deux raisons majeures de procéder à cette installation manuelle : d'une part, obtenir un programme PHP compilé à l'intérieur même de l'objet compilé Apache et y gagner sensiblement en performances ; d'autre part, disposer des répertoires concernés par PHP à l'intérieur du site de façon à les faire bénéficier des précautions de sécurité indispensables.

Voir, en français, les passages du manuel qui s'inspirent de ce guide :

- les composants concernés ;
- les étapes de l'installation ;
- la préparation des sources, leur assemblage, leur compilation :

  http://www.php.net/manual/fr/install.apache.php

Consultez, de toute façon, le document install.txt placé dans le répertoire PHP. Il constituera l'information la plus récente sur le produit que vous téléchargez et sur son installation en contexte Windows.

Installation dans un package easyphp

La facilité d'installation et d'utilisation de PHP, sa prédominance dans le monde du logiciel libre, le succès parallèle d'Apache et de MySQL ont conduit des développeurs à présenter un « package » englobant les trois éléments : Apache, PHP et MySQL, à destination de la plate-forme Windows, en offrant de l'installer « en un tour de main ». Et le package est gratuit !

Prêt à l'emploi

Solution rêvée pour tous ceux qui se proposent de s'initier à ces composants sur leur PC, en prévision de l'installation de productions plus complexes, mais aussi pour les très nombreuses personnes qui se proposent de monter un site à peu de frais et disposent ainsi d'emblée de tout l'outillage pour le construire. Soit qu'ils le confient à un fournisseur d'accès, et tous accueillent Apache, PHP et MySQL. Soit qu'ils installent serveur et site sur leur propre PC et en gèrent eux-mêmes les accès.

Cette population s'accroît rapidement et sans doute, cher lecteur, êtes-vous au nombre de ces dynamiques entrepreneurs ! À ce jour, le site enregistre plus de 200 000 téléchargements pour les versions antérieures à la 1.6 et plus de 340 000 pour la version courante !

Un site officiel est dédié au package easyphp :

www.easyphp.org

Une installation « cool »

La page d'accueil vous permet de faire connaissance avec tous les services associés. Au besoin, modifiez la langue d'affichage.

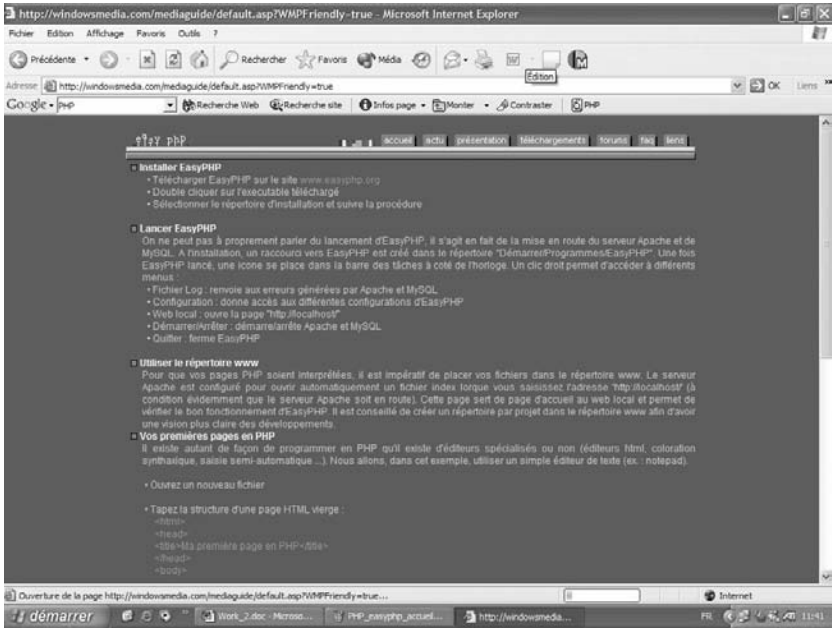
Vérification : vous avez sous les yeux un site servi par Apache. Observez dans l'adresse de votre *browser* :

- le suffixe .php3 associé à la page demandée (plus exactement au script appelé) ;
- la page envoyée par défaut appelée « index.php3 » ;
- le paramètre de « lang= » que votre *browser* envoie au module de négociation du serveur après que vous ayez demandé un langue particulière ;
- la structure du site : quel que soit l'onglet demandé, tous les documents se trouvent directement sous *www.easyphp.org/*, c'est-à-dire tous au même niveau, sous la racine, et donc sous htdocs.

Notez que l'équipe du projet procède à ses propres tests de compatibilité entre composants, de performances et de sécurité. Le cocktail présenté ne comporte pas forcément la dernière version de chaque composant, mais

une combinaison que l'équipe présente comme opérationnelle. Ainsi, la version 1.6 comporte :

- Apache 1.3.24
- PHP 4.2.0
- MySQL 3.23.49
- phpmyadmin 2.2.6 (utilitaire d'administration)



▸ Rendez-vous à l'onglet de téléchargement.

▸ Choisissez la version courante. A ce jour, il s'agit de easyphp 1.6, d'une taille de 10 584 ko. Dans l'onglet de présentation, vous trouvez le mode d'emploi.

▸ Un double clic sur le fichier téléchargé et ensuite laissez-vous conduire.

Acceptez les termes de la licence. Essentiellement, ils spécifient que l'utilisation du produit est libre, mais que vous ne pouvez pas prétendre l'avoir développé vous-même !

L'installation se termine : confirmez que vous souhaitez ouvrir la page d'accueil et lancer easyphp.exe.

Déjà, le démarrage

En dehors de ce contexte d'installation, vous lancez easyphp comme tout autre programme, via le menu *Démarrez / Programmes / EasyPHP*.

Vous verrez s'ouvrir, quelques courts instants, la fenêtre dos qui signale le lancement d'Apache.

Et, puisque cette installation fut si rapide et si simple, poursuivez en fabriquant la première page, comme proposé dans le document « Présentation ». Cet exemple est manifestement le jumeau de celui que nous proposions en début de ce chapitre, et qui affichait un sympathique « Bonjour, la compagnie ».

Pour respecter les habitudes européennes de format de date, vous pouvez aussi rédiger : print date (" D d M Y ") ; ce qui donne dans l'ordre : le jour dans la semaine, le jour dans le mois, le mois et l'année. En l'occurrence l'instruction echo est équivalente au print suggéré.

Des répertoires aménagés

Soyez attentif au choix du répertoire. C'est bien sous le répertoire Program Files / Easyphp / www que les concepteurs du package proposent de placer le texte de l'exemple. C'est une autre formule que celle proposée par défaut en installation du seul serveur Apache : le répertoire par défaut y était Apache2/htdocs. C'est donc là que se trouve votre site et que vous pouvez progressivement placer les documents que vous mettrez à disposition de vos clients.

Remarquez d'ailleurs que votre document se retrouve là en compagnie d'un autre : index.php. Et qui sera le document par défaut si vous n'en spécifiez aucun dans votre appel à *http://127.0.0.1/*, ou, évidemment, si vous demandez *http://127.0.0.1/index.php*. Là, vous êtes bien dans le site.

Il existe un autre fichier index.php dans la structure, mais dans le répertoire /home, en compagnie d'autres documents qui constituent la présentation du site easyphp. Là, ce n'est plus à proprement parler le site, mais une structure de fichiers de commentaires et d'assistance. Vous accédez par *http://127.0.0.1/home/index.php* ou, plus simplement, par le menu « Administration » lorsque vous cliquez droit sur la petite icône « E. » dans la barre des tâches, dans le coin inférieur droit de votre écran.

► Ne résistons pas au plaisir de vérifier comment le serveur Apache a été configuré dans cette installation. Rendez-vous en Apache (et non Apache2, puisqu'il s'agit de Apache 1.3), au centre de pilotage : Apache/conf/httpd.conf.

► La directive ServerName nous indique « localhost ». C'est l'équivalent de 127.0.0.1, l'adresse TCP/IP qui désigne la machine locale. Vous pouvez donc interroger votre site en introduisant dans l'adresse de votre *browser* : *http://localhost/*. Inutile de mentionner le port, puisque la directive Port donne la valeur par défaut : 80.

► Pour faire fonctionner votre premier script PHP que vous aurez sans doute nommé Date.php, vous demandez indifféremment *http://localhost/Date.php* ou *http://127.0.0.1/Date.php*.

Et maintenant, bienvenue à la muse. Il n'y a plus qu'à élaborer votre site ! Il n'y a qu'à placer des documents et des scripts PHP dans le répertoire du site www. Il n'y a qu'à...

Installation en contexte Linux

Comme en contexte Windows, vous pouvez choisir entre les deux formules d'association de PHP à Apache : soit en tant de programme CGI extérieur, soit en tant que module imbriqué dans l'objet général compilé, la seconde formule comportant les avantages qui ont été évoqués plus haut.

Par ailleurs, vous pouvez aussi choisir de partir du programme déjà compilé, le « *binary* », ou du code source que vous compilerez vous-même.

Vous aurez aussi à modifier le fichier de configuration d'Apache, le httpd.conf, pour y introduire les références au nouveau venu PHP.

Voici les indications de décompactage, d'installation et de configuration, extraites du manuel, pour une installation sur Apache 1.3. À ce jour, l'installation de PHP sur Apache 2.0 est toujours considérée comme soumise aux tests, et n'est pas encore recommandée pour les serveurs de production sensibles.

1. gunzip apache_1.3.x.tar.gz
2. tar xvf apache_1.3.x.tar
3. gunzip php-x.x.x.tar.gz
4. tar xvf php-x.x.x.tar
5. cd apache_1.3.x
6. ./configure—prefix=/www
7. cd ../php-x.x.x
8. ./configure—with-mysql—with-apache=../apache_1.3.x—enable-track-vars
9. make
10. make install
11. cd ../apache_1.3.x

12. for PHP 3: ./configure—activate-module=src/modules/php3/libphp3.a
for PHP 4: ./configure—activate-module=src/modules/php4/libphp4.a
13. make
14. make install

- Au lieu de cette étape, vous pouvez aussi copier le binaire httpd et remplacer votre exécutable actuel. Assurez-vous tout de même que le serveur est bien désactivé.

15. cd ../php-x.x.x
16. pour PHP 3: cp php3.ini-dist /usr/local/lib/php3.ini
    pour PHP 4: cp php.ini-dist /usr/local/lib/php.ini

- Vous pouvez éditer votre fichier php.ini pour modifier certaines options PHP. Si vous préférez placer ce fichier ailleurs, utilisez : with-config-file-path=/path lors de l'étape 8.

17. Éditez votre fichier httpd.conf ou srm.conf file et ajoutez :
Pour PHP 3: AddType application/x-httpd-php3 .php3
Pour PHP 4: AddType application/x-httpd-php .php

- Vous pouvez choisir n'importe quelle extension que vous voulez ici. .php est uniquement une suggestion. Vous pouvez aussi inclure .html.

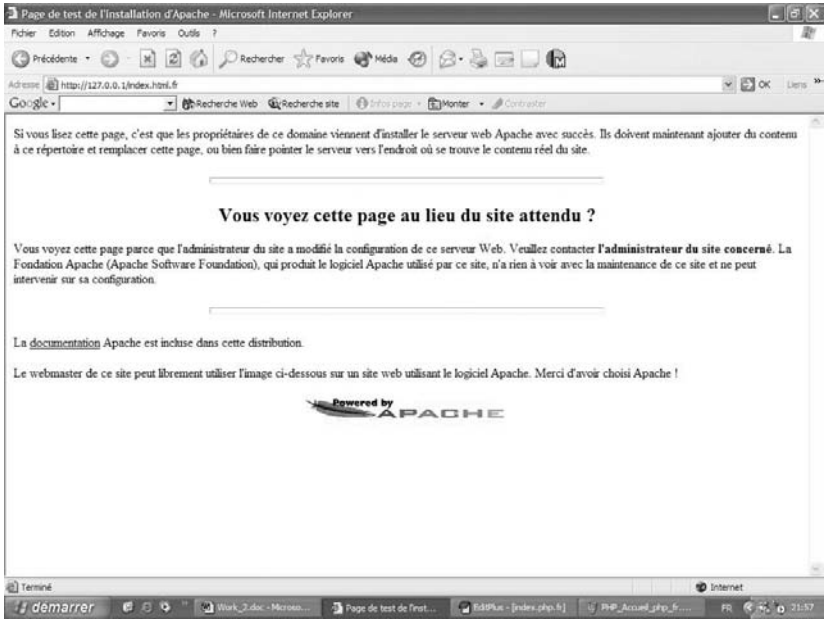
18. Utilisez votre procédure habituelle pour démarrer votre serveur Apache. (Vous devez l'éteindre et le redémarrer, pas seulement lui envoyer un signal HUP ou USR1.)

Fonctionnement de PHP

Pour vérifier le bon fonctionnement de votre installation PHP, nous allons procéder à deux tests très simples.

Une page personnalisée

Le premier test va mettre en exergue que les scripts PHP sont bien encastrés dans des pages HTML. Prenons donc une page .html que nous connaissons bien, pour l'avoir déjà fait fonctionner.


Page de test de l'installation d'Apache - Microsoft Internet Explorer
Si vous lisez cette page, c'est que les propriétaires de ce domaine viennent d'installer le serveur web Apache avec succès. Ils doivent maintenant ajouter du contenu à ce répertoire et remplacer cette page, ou bien faire pointer le serveur vers l'endroit où se trouve le contenu réel du site.
Vous voyez cette page au lieu du site attendu ?
Vous voyez cette page parce que l'administrateur du site a modifié la configuration de ce serveur Web. Veuillez contacter l'administrateur du site concerné. La Fondation Apache (Apache Software Foundation), qui produit le logiciel Apache utilisé par ce site, n'a rien à voir avec la maintenance de ce site et ne peut intervenir sur sa configuration.
La documentation Apache est incluse dans cette distribution.
Le webmaster de ce site peut librement utiliser l'image ci-dessous sur un site web utilisant le logiciel Apache. Merci d'avoir choisi Apache !
Powered by APACHE

Lorsque, depuis votre *browser*, vous demandez le site que vous avez installé sur la même machine, vous obtenez une page d'accueil qui doit ressembler à la capture écran ci-contre.

Nous savons :

- qu'elle se trouve dans le répertoire par défaut du site, c'est-à-dire : C: \Program Files \Apache Group\Apache2\htdocs ;
- et que, parmi les divers documents index.html disponibles, c'est l'index.html.fr qui vous est livré parce que le *browser* aura demandé à communiquer en français. Ainsi, vous aurez obtenu cette page d'accueil en introduisant *http://127.0.0.1*.

Le *browser* aura suppléé en ajoutant la barre oblique finale : il imagine que vous demandez ce qui se trouve à la racine du site. Puisque le document livré par défaut sera index.html.fr, vous auriez tout aussi bien pu introduire *http://127.0.0.1/index.html.fr*.

Je vous invite d'ailleurs à l'expérimenter par vous-même.

Une variante à tester

Puisqu'un script PHP est immergé dans une page HTML et générera « quelque chose » à l'intérieur de cette page, éditons-la pour y insérer un script avec un éditeur ASCII quelconque, tel Notepad. Par prudence, effectuons une copie de la page, et éditons-la.

La commande PHP que nous allons exploiter est echo ; elle affiche dans le document à livrer un contenu qu'on lui soumet. Le script PHP sera reconnu par l'analyseur de texte grâce aux balises conventionnelles de PHP : au début < ? ou < ?php, et en finale ?>.

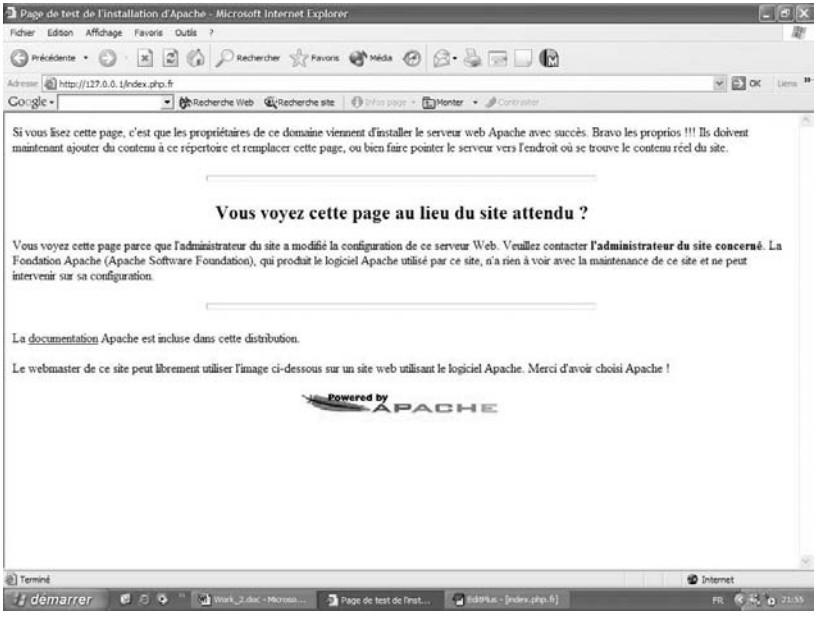
Je vous propose l'insertion suivante, mais laissez-vous inspirer par votre muse :

```
<p>Si vous lisez cette page, c'est que les propriétaires de ce
domaine viennent d'installer le
serveur web Apache
avec succès.
<? echo "Bravo les proprios !!! "; ?>
Ils doivent maintenant ajouter du contenu à ce
répertoire et remplacer cette page, ou bien faire pointer le serveur
vers l'endroit où se trouve le contenu réel du site.</p>
```

Attention lorsqu'il s'agit de sauver cette nouvelle version de la page d'accueil. C'est bien dans le répertoire htdocs que nous la sauverons puisque c'est celui-ci qui est mentionné dans le php.ini comme le dépositaire des scripts PHP. Mais, conformément aux indications rajoutées dans le httpd.conf, Apache ne demandera la collaboration de PHP que s'il trouve des documents qui sont suffixés par .php.

Sauvons donc notre nouvelle version en C:\Program Files\Apache Group\Apache2\htdocs sous le nom de index.php.fr.

Relancez votre *browser* en explicitant la demande de ce nouveau document : *http://127.0.0.1/index.php.fr*.


Page de test de l'installation d'Apache - Microsoft Internet Explorer
Si vous lisez cette page, c'est que les propriétaires de ce domaine viennent d'installer le serveur web Apache avec succès. Bravo les proprios !!! Ils doivent maintenant ajouter du contenu à ce répertoire et remplacer cette page, ou bien faire pointer le serveur vers l'endroit où se trouve le contenu réel du site.
Vous voyez cette page au lieu du site attendu ?
Vous voyez cette page parce que l'administrateur du site a modifié la configuration de ce serveur Web. Veuillez contacter l'administrateur du site concerné. La Fondation Apache (Apache Software Foundation), qui produit le logiciel Apache utilisé par ce site, n'a rien à voir avec la maintenance de ce site et ne peut intervenir sur sa configuration.
La documentation Apache est incluse dans cette distribution.
Le webmaster de ce site peut librement utiliser l'image ci-dessous sur un site web utilisant le logiciel Apache. Merci d'avoir choisi Apache !
Powered by APACHE
démarrer

Et, cela ne devrait pas être une surprise, mais tout de même une satisfaction, vous obtenez la page ci-contre.

Dans ce cas, Apache n'a pas seulement extrait une page statique du site pour la renvoyer au client, mais il a fait appel au moteur ZEND de PHP pour qu'il analyse un script et réalise la commande (echo) qui lui était soumise. PHP a produit une version complétée du document, en y insérant son propre résultat.

Des lendemains qui chantent

Vous devinez que, grâce à son langage complet, vous aurez la possibilité de déclencher dans votre site des programmes très puissants et de restituer à vos clients des résultats personnalisés, et aussi à jour et copieux que souhaité.

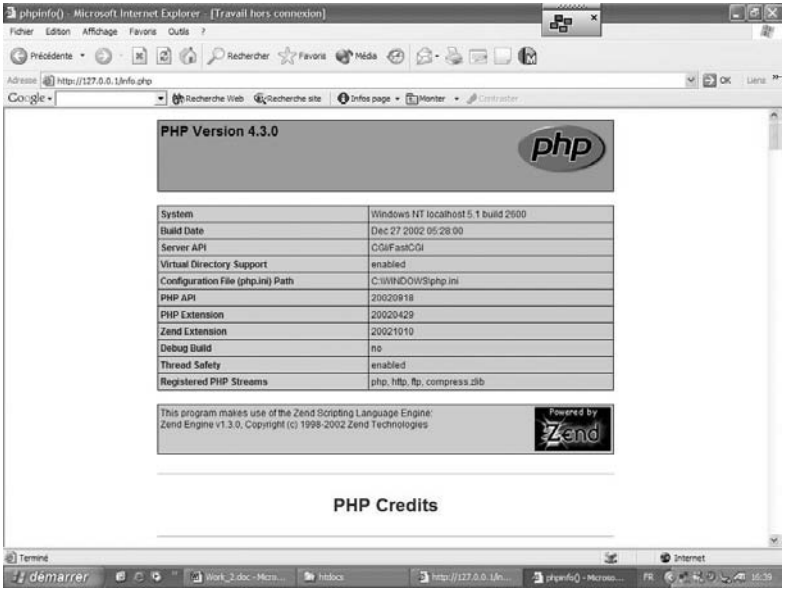
Une demande simple pour une page utile

Le second test fera appel à une autre commande de PHP phpinfo() : elle livre de nombreuses informations relatives à la configuration de PHP et du serveur qui le supporte.

- Créons un nouveau fichier.
- Introduisons le texte suivant : `< ? phpinfo() ; ?>`

C'est un texte au régime light, qui ne comporte même pas d'encadrement HTML ; mais c'est suffisant pour que PHP en fabrique la longue page HTML que vous allez voir apparaître à l'écran.

- Sauvons-le dans le répertoire du site htdocs, par exemple sous le nom de info.php, et relançons notre *browser* avec la demande : *http://127.0.0.1/info.php*.



Vous recevez bien ceci comme un bilan de santé de votre système : des indications sur Apache, sur PHP et même sur la demande ponctuelle que vous venez d'introduire. Le test est donc réussi et son résultat va nous permettre de faire plus ample connaissance avec PHP.

Configurer le noyau de PHP

La page d'informations livrée par le script de phpinfo() va nous guider pour vérifier ce que nous avons prévu comme configuration pour le noyau de PHP. Elle nous aidera bientôt à situer les extensions possibles à ce noyau, car ce sont elles qui comporteront les fonctions les plus recherchées.

La page d'introduction nous rappelle la version implantée de PHP, le système d'exploitation de la machine-hôte, la localisation que nous avions choisie pour le fichier php.ini. Elle mentionne aussi l'existence, en principe indépendante, du moteur d'interprétation et d'exploitation ZEND, et son numéro de version.

Passons à présent en revue quelques-uns des paramètres du noyau...

Les options de langage

Engine = on — on/off active ou désactive l'analyseur PHP, ce qui peut se révéler utile en session de maintenance.

Short_open_tag	accepte < ? comme balise début plutôt que < ?php.
Asp_tags	on : les balises <% %> du type du serveur ASP seront aussi reconnues.
Safe_mode	on : des contrôles de sécurité supplémentaires sont activés, comme par exemple, la vérification que tel fichier ou tel répertoire n'est accessible que s'il a le même propriétaire que le script. Le paramètre safe_mod_exe_dir mentionne les répertoires des programmes qui sont soumis au mode sécurisé.

Le suivi des erreurs et la journalisation

Error_reporting	Permet de combiner des valeurs indiquant le niveau d'erreur à afficher, depuis les erreurs proprement dites jusqu'aux alertes et aux simples notes de l'utilisateur.
Display_errors	Permet d'obtenir que les messages d'erreur soient redirigés vers la console.
Log_errors	Permet de diriger l'affichage des erreurs vers le journal standard du serveur ou vers une fichier dédié aux erreurs PHP et désigné par error_log.

Le paramétrage de la gestion des données

Les options de gestion de données portent notamment sur la manière de traiter les caractères spéciaux. Exemple : « la grotte de MarieGobaye ». Cette chaîne de caractères comporte des guillemets. Si elle était passée au gestionnaire de base de données MySql, celui-ci considérerait que derrière le dernier guillemet, se trouverait... autre chose, et pour lui ce serait un ordre SQL. Ce qui conduirait à une erreur d'exploitation.

Les options permettent d'introduire ce qu'on appelle des caractères d'échappement, tels que la barre oblique, de reconnaître un guillemet comme un caractère spécial, d'insérer automatiquement cette barre oblique.

Les extensions à PHP

Après avoir installé PHP, vous souhaiterez certainement ajouter une série d'extensions d'emblée disponibles.

- Un coup d'œil d'abord sur le répertoire système C: \Windows, à la recherche des programmes mêmes. Recherchons-y les fichiers commençant par php et suffixés par .dll. À ce jour, on y trouve par exemple, placés lors de l'installation de PHP, php_imap.dll et php_calendar.dll qui assurent des fonctions de messagerie (IMAP) et de calendrier. Mais aussi un fichier php4ts.dll qui comprend un grand nombre de programmes disponibles.

  Les extensions sont nommées dans le fichier php.ini, mais en commentaire du fait du point-virgule qui les précède.

  Il suffit de retirer le point-virgule pour rendre une extension active. Ainsi, devant ; extension=php_imap.dll.

Mais vous cherchez sans doute des vedettes telles que MySql ou ODBC, et ces extensions ne figurent pas dans la liste. Effectivement ! C'est que les développeurs de PHP sont attentifs aux besoins et aux habitudes des utilisateurs, et depuis PHP 4.0.5, ils ont d'office rendu disponibles les extensions MySql, ODBC, FTP, Calendar, BCMath, COM, PCRE, Session, WDDL et XML.

- Voyez à l'aise la liste des extensions présentée en php.ini.

```
;Windows Extensions
;Note that MySQL and ODBC support is now built in, so no dll is needed for it.
```

Certains programmes extensions peuvent recevoir des paramètres. Une section de php.ini permet de garnir des variables à cet effet. Ne prenons que l'exemple d'ODBC, avec ses trois premiers paramètres.

```
[ODBC]
; Allow or prevent persistent links.
odbc.allow_persistent = On
; Check that a connection is still valid before reuse.
odbc.check_persistent = On
; Maximum number of persistent links. -1 means no limit.
```

odbc.max_persistent = -1

; Maximum number of links (persistent + non-persistent). -1 means no limit.

odbc.max_links = -1

Uodbc.allow_persistent Si cette option est positionnée à 'on', les connexions permanentes sont possibles.

Uodbc.max_persistent Nombre maximal de connexions persistantes ; -1 signifie " à l'infini "

Uodbc.max_links Nombre maximal de connexions permanentes et non permanentes ; -1 signifie " à l'infini ".

La gestion de session est aussi une fonctionnalité appréciée car elle supporte le dialogue client-serveur sous forme de transaction.

Mais encore...

Les ouvrages sur PHP ou sur le couple PHP – MySQL se sont multipliés récemment, pour débutants comme pour experts.

Vous en trouverez aisément les références sur les sites des maisons d'édition, dont votre préférée... *www.editions-eyrolles.com* !

J'en épingle un parce qu'il présente, en plus des éléments du langage PHP, l'exemple d'une construction de site commercial interactif : le site des éditions Eyrolles. L'intérêt de PHP est démontré par l'aboutissement du projet de site, celui aussi de la triade Linux/Apache/PHP.

En documentant la démarche de conception et de réalisation du site, les auteurs guident le lecteur au long du processus de mise au point d'un site opérationnel. Par ailleurs, plutôt que d'énumérer les caractéristiques du langage, ils ont préféré « analyser » ce langage selon plusieurs axes : le positionnement de PHP sur le marché, la description évolutive des fonctions et librairies disponibles, le mise en œuvre d'un cas réel et enfin les possibilités de capitalisation autour du langage.

- Laurent Lacroix, Nicolas Leprince, Christophe Boggero, Christophe Lauer, *Programmation Web avec PHP*, Eyrolles, Deuxième tirage 2000, 365 pages.
- À mentionner aussi G.A. Leierer, R. Stoll, *PHP & MySQL*, Micro Application, Grand Livre, 2001. Clarté dans la présentation des fonctions de PHP et MySQL, et présentation d'un cas de réalisation de site. Orienté vers la programmation de bases de données pour Internet, et donc de sites Web interactifs.

Il existe de nombreux sites qui traitent de PHP. Une simple demande sur le mot clé PHP sur un site chercheur ou méta-chercheur vous en donne à profusion. Pas moins de 183 millions de références, via Google, à ce jour ! Plus de 500 000 références comportant le mot clé dans les adresses ! Reconnaissance aux systèmes de recherche qui tentent de trier les résultats selon un intérêt présumé !

Vous y observerez que des sites centraux ont leurs antennes dans différents pays. Mais les références incontournables sont celles du site officiel du projet PHP et des sites apparentés :

- Le site du projet *www.php.net*, où vous conduit aussi le site *www.apache.com*. Sous ce site :
  - des news,
  - des FAQ,
  - des « tutoriels »,
  - le manuel complet.
- *www.e-gineer.com/phpkb/* référence à suivre régulièrement pour ses liens tenus à jour, l'abondance et la qualité des contributions.
- *www.easyphp.org* fournit, sous l'onglet *liens,* une liste des sites officiels apparentés : PHP, ZEND, MySQL, PhpMyAdmin ainsi que les sites de documentation.
- Tout comme la plupart des autres réalisations *OpenSource*, le manuel proposé sur le site de *www.php.net* est le fruit de la collaboration de nombreux participants. Vous y noterez le nom du concepteur du langage : Rasmus Lerdorf.